**Strange order**

Given an integer n . Initially you have permutation p of size n : p[i] = i . To build new array a from p following steps are done while permutation p is not empty:

Choose maximum existing element in p and define it as x ; Choose all such y in p that gcd ( x , y ) ≠ 1 ; Add all chosen numbers into array a in decreasing order and remove them from permutation. Your task is to find a after p became empty.

Note: gcd ( a , b ) is the greatest common divisor of integers a and b .

#include<bits/stdc++.h>

#include<iostream>

#include<vector>

#include<algorithm>

using namespace std;

int main() {

int n;

cin>>n;

vector<int> firstFact(n + 1, 0);

firstFact[1] = 1;

for(int i = 2; i <= n; i++){

if(firstFact[i] == 0){

for(int j = 1; i \* j <= n; j++){

if(firstFact[i \* j] == 0)

firstFact[i \* j] = i;

}

}

}

vector<int> temp;

for(int i = n; i > 0; i--){

if(firstFact[i] == 0)

continue;

cout<<i<<" ";

int t = firstFact[i], t1 = i, c;

firstFact[i] = 0;

while(t1 >= 1){

c = i;

while(c - t > 1 & t != 1){

if(firstFact[c - t])

temp.push\_back(c - t);

firstFact[c - t] = 0;

c = c - t;

}

if(t1 == t){

sort(temp.begin(), temp.end());

for(int i = temp.size() - 1; i >= 0; i--)

cout<<temp[i]<<" ";

//cout<<endl;

temp.clear();

break;

}

while(t1 % t == 0)

t1 = t1 / t;

t = firstFact[t1];

}

}

return 0;

}